RECEIVED
CENTRAL FAX CENTER

AUG 0 8 2006

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s)	:	Hausman, Steven Michael
Application #	:	10/622,259
Confirmation #	:	3269
Filed	:	18 July 2003
Application Title	:	Automatic Configuration of a Remote Modem
Art Unit #	:	2182
Latest Examiner	:	Hassan, Aurangzeb

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION UNDER 37 C.F.R. § 1.132

Sir:

I, Dr. Ronald D. Williams, a citizen of the United States, whose full post office address is 1715 Hearthglow Lane, Charlottesville, VA 22901 declare as follows under penalty of perjury.

Background

1. I hold a Ph.D. degree in Electrical Engineering from the Massachusetts Institute of Technology awarded in 1984.
2. I hold a M.S. degree in Electrical Engineering from the University of Virginia awarded in 1978.
3. I hold a B.S. degree in Electrical Engineering from the University of Virginia

awarded in 1977.

4. I am currently an associate professor of Electrical & Computer Engineering at the University of Virginia.
5. Since 1984, I have worked continually in the field of electrical engineering with particular emphasis in embedded computing with applications in control and signal processing.
6. During my career, I have been granted five U.S. patents for my own inventions in the field of embedded computing.

Review

7. I have reviewed Application Serial No. 10/622,259 (hereinafter the present application).
8. I know what one of ordinary skill in the art of the present application would have known on the priority date claimed by the present application (23 December 2002).
9. I have reviewed the USPTO Office Action dated 7 June 2006 (hereinafter the "Office Action") regarding Application Serial No. 10/622,259.
10. I have reviewed U.S. Patent No. 5,649,001 (Thomas).
11. I have reviewed pages 3, 119, and 297 of the book *REAL-TIME SYSTEMS*, Wolfgang A. Halang, Krzystztof M. Sacha, World Scientific, 1993 ("Halang").
12. I have reviewed page 42 of the book *REAL-TIME SYSTEMS Scheduling, Analysis and Verification*, Albert M.K. Cheng, Wiley Interscience, Aug. 12, 2002 ("Cheng").

13. Among the devices with which I was familiar prior to 23 December 2002, the priority date claimed by the present application, were devices of the type recited in Thomas.

Lexicography

14. Each of claims 1-32 recites a "programmable logic controller" ("PLC"). One having ordinary skill in the art as of 23 December 2002, the priority date of the present application, would construe the phrase "programmable logic controller" to mean:

> a solid-state, microprocessor-based, hard real-time computing system that is used, via a network, to automatically monitor the status of field-connected sensor inputs, and automatically control communicatively-coupled devices of a controlled industrial system (e.g., actuators, solenoids, relays, switches, motor starters, speed drives (e.g., variable frequency drives, silicon-controlled rectifiers, etc.), pilot lights, ignitors, etc.) according to a user-created set of values and user-created logic and/or instructions stored in memory. The sensor inputs reflect measurements and/or status information related to the controlled industrial system. A PLC provides any of: automated input/output control; switching; counting; arithmetic operations; complex data manipulation; logic; timing; sequencing; communication; data file manipulation; control; relay control; motion control; process control; distributed control; and/or monitoring of processes, manufacturing equipment, and/or other automation of the

controlled industrial system. In addition to controlling a process, a PLC might also provide control of information, such as via outputting information to speakers, printers, monitors, displays, indicators, etc., and/or rendering information, such as via reports, notifications, and/or alarms, etc., such as via a Human-Machine Interface (HMI). Because of its precise and hard real-time timing and sequencing capabilities, a PLC is programmed using ladder logic or some form of structured programming language specified in IEC 61131-3, namely, FBD (Function Block Diagram), LD (Ladder Diagram), ST (Structured Text language), IL (Instruction List) and/or SFC (Sequential Function Chart), or potentially via a general purpose hard-real-time-aware programming language, such as ADA. Because of its hard real-time timing and sequencing capabilities, a PLC can replace up to thousands of relays and cam timers. PLC hardware often has good redundancy and fail-over capabilities.

15. One having ordinary skill in the art as of 23 December 2002, the priority date of the present application, would have found implicit support in the present application for this definition at least at paragraph 0002 of the published version of the present application (U.S. Patent Publication No. 20050261026).

16. One having ordinary skill in the art as of 23 December 2002, the priority date of the present application, would have found support for the definition of paragraph 14 in prior art publications. For example, support for the definition of paragraph 14 can be

found in Halang at least at pages 3 and 297.

17. One having ordinary skill in the art as of 23 December 2002, the priority date of the present application, would have construed the phrase "hard real-time", as recited in the definition of paragraph 14 to mean:

> relating to a system (or sub-system) having activities with hard deadlines, and a sequencing goal of always meeting all those hard deadlines. A system operating in non-real-time can suffer a critical failure if time constraints are violated. A classic example of a real-time computing system is an automobile engine electronic valve timing control system, in which an overly delayed or overly advanced control signal might cause engine failure or damage, due to one or more valve-piston collisions. Systems operating in real-time typically utilize instructions embedded in hardware and/or firmware.

18. One having ordinary skill in the art as of 23 December 2002, the priority date of the present application, would have found implicit support in the present application for this definition at least at paragraph 0002 of the published version of the present application (U.S. Patent Publication No. 20050261026).

19. One having ordinary skill in the art as of 23 December 2002, the priority date of the present application, would have found support for the definition of paragraph 17 in prior art publications. For example, support for the definition of paragraph 17 can be found in Halang at least at page 3 and in Cheng at least at page 42.

20. One having ordinary skill in the art as of 23 December 2002, the priority date of the present application, would have understood the phrase "hard deadline" comprised in the definition of "real-time" (see paragraph 17, *supra*) to mean:

> a special case where completing an activity within the deadline results in a system receiving all the utility possible from that activity, and completing the activity outside of the deadline results in zero utility (i.e., resources consumed by the activity were wasted, such as when one travels to the beach to photograph a sunrise on a particular day and arrives after the sun has already arisen) or some negative value of utility (i.e., the activity was counter-productive, such as when firefighters enter a burning building to search for a missing person seconds before the building collapses, resulting in injury or death to the firefighters). The scheduling criteria for a hard deadline is to always meet the hard deadline, even if it means changing the activity to do so.

21. One having ordinary skill in the art as of 23 December 2002, the priority date of the present application, would have found implicit support in the present application for this definition at least at paragraph 0002 of the published version of the present application (U.S. Patent Publication No. 20050261026).

22. One having ordinary skill in the art as of 23 December 2002, the priority date of the present application, would have found support for the definition of paragraph 20 in prior art publications. For example, support for the definition of paragraph 20 can be

Table 1.1: Hard and soft real-time environments

Real-time environment	Hard	Soft
Consequences of not responding in time	useless results, or danger for people or the process	costs increasing with waiting time
Examples	car driver, housewife power plant flight guidance system steel mill	beginning of a party airline reservation bank library system
Analysis objectives	worst case performance	average performance

Despite the similarity, the last two examples differ significantly from the others. A driver, a housewife, and a power plant control system operate in very rigid environments. They must always respond in time and cannot cease operation even for a moment. The reservation and bank supporting systems work in much more flexible environments. The computer should on average respond quickly, but occasional delays are acceptable. Moreover, in the case of a failure, a temporal suspension of the system operation may be tolerated provided that the data recorded in a database are preserved. The difference is fundamental. It may be expressed by referring, respectively, to hard and soft real-time conditions. The bank and airline reservation systems are examples of soft real-time business applications where the problem of data integrity has priority over the problem of timeliness. The power plant control system is an example of a hard real-time process control application where the problem of timeliness is prevalent. Hard and soft real-time systems can be distinguished by the consequences of violating the timeliness requirement: whereas soft real-time environments are characterized by costs rising with increasing lateness of results, such lateness is not permitted under any circumstances in hard real-time environments, because late computer reactions are either useless or even dangerous. In other words, the costs for missing deadlines in hard real-time environments are infinitely high. Hard real-time constraints can be determined precisely, and typically result from the physical laws governing the technical processes being controlled [2]. Short characteristics of environments and systems of both types are assembled in Table 1.1.

This book focuses on hard real-time industrial process control and discrete manufacturing systems. Soft real-time systems are not discussed, although the methods and tools described herein can be applied to address problems of this type. The book is organized as follows. Chapter 1 provides the reader with introductory information concerning problems, structures, and historical perspective of industrial real-time systems. The conceptual foundations are presented in Chapters 2 and 3. The next three Chapters: 4, 5, and 6, deal with hardware architecture problems and solutions. Chapters 7 and 8 describe principles, basic elements, and examples of oper-

The standards are fully compatible and the only difference between them is at the lowest layers (physical and medium access control). The reason for this diversity lies in the different requirements for factory floor and office networks.

The main issues in designing communication systems for distributed real-time applications are those of high reliability and of deterministic response times under heavy load. These demands can be related to the physical structure of the network in the form of the following three requirements:

- There should be no distinguished network master node.
- A break-down of a single node cannot affect the communicational abilities of other nodes.
- Each node must have guaranteed access to the transmission line in a deterministic time.

Moreover, the network should handle a large number of relatively short messages efficiently. The requirements in an office automation system are for high average throughput and efficient handling of relatively large data files. For these reasons, TOP uses the ISO 8802.3 CSMA/CD bus specification and MAP uses the ISO 8802.4 token passing bus specification for physical and medium access control layers.

MAP architecture. The physical layer specification of MAP provides two alternatives: 10 Mbps broadband bus using three-level duo-binary amplitude modulation with phase shift keying for backbone networks, and 5 Mbps carrierband bus using frequency shift keying for small subnets. The transmission medium is consequently 75 Ω coaxial cable. The broadband version uses a single cable with a head-end remodulator, and needs two transmission channels each with 12 MHz bandwidth. The carrierband version uses neither amplifiers nor head-end remodulator, which makes the bus bidirectional and entirely passive.

The lower half of the data link layer (medium access control) implements the token passing scheme. The upper half of the data link layer (logical link control) is specified to provide connectionless datagram service. This speeds the frame transfer up, but provides no flow control and no error recovery functions. The LLC protocol can be used to link together networks which differ at physical and MAC levels, e.g., token passing subnetworks with different tokens.

The network layer is responsible for routing messages between network users, and becomes important if the messages have to be sent between different subnetworks interconnected within a wider network. For reasons of speed, the network layer of MAP is defined by the ISO 8473 connectionless Internet protocol. Internetworkability with connection-oriented networks, particularly with TOP, is achieved by means of

described in Chapter 5. Basic characteristics of a sampled family of typical PLCs are given in Table III.1.

Usually, PLCs use neither operator terminals nor mass memories, such as disks or tapes, which are sensitive to the harsh industrial environment. The basic operating software resides entirely in main memory, and a PLC operates autonomously without human intervention.

The main memory of a PLC is divided into specialized segments designated for different purposes. Depending on the intended use, the segments can be organized in bits or words. Basic segments contain the following types of information:

Operating system. The segment is organized in words and contains the code and the data of the operating system; the system code is usually stored in a ROM memory, while the data are stored in a RAM segment with battery backup, which is carefully protected against incursion from the other programs.

Application programs. The segment is organized in words and contains the instructions of application programs.

Application data. The segment is subdivided into three areas containing the variables into which the machine puts sensor data, the intermediate variables, and the output variables associated with the actuators. PLCs offering only logic functions have the data segment organized in bits, while the more advanced machines have part of this area made up of words.

There is a strict correspondence between the positions of the bits of the input and output data areas and the layout of input and output signals at the connectors and cables leading in and out the PLC. All the input-output operations are implicit on PLCs. This means that the application programs refer to the data areas in memory only, while the physical data transfer is handled entirely by the system software or firmware. In most cases, all input and output data are transferred in and out the machine at regular time intervals through direct memory access (DMA) channels.

The processor is the operational unit designated to control the machine as a whole and to execute instructions of the programs. The instructions are fetched from consecutive words of the program segments of the main memory, which are addressed by the program counter. The data (variables) are addressed in the data segment of the main memory using a variety of addressing modes. The operations of the instructions take place in the accumulator registers. Partial results of the operations are stored on a stack, which constitutes an internal memory of the processor. The processors intended for small controllers contain only 1-bit accumulators and can execute operations only on bits. More advanced processors contain 8 . . . 32-bits accumulators and can execute operations on bits or words.

The basic instruction set of a PLC processor contains the instructions for loading and storing data from and to the memory, logical operations, and conditional and